a plurality of peripheral devices that accept the processor requests; and

a controller responsive to the processor requests that creates a plurality of separate pending queues corresponding to each one of the plurality of peripheral devices for queuing the processor requests directed to a particular peripheral device in entries of a corresponding separate pending queue, wherein at least two separate peripheral devices process the processor requests simultaneously, after retrieving such processor requests from their respective separate pending queues.

40. (New claim) The data processing system according to claim 39, wherein the one or more requesting processors include dependency checking logic that generate non-blocking processor requests.

41. (New claim) The data processing system according to claim 40, wherein the non-blocking processor requests are generated by running real-time processes.

42. (New claim) The data processing system according to claim 39, wherein the processor requests are prioritized in the pending queues such that the higher priority processor requests are processed before the lower priority processor requests.

4 ~~43~~. (New claim) The data processing system according to claim 1 ~~39~~, wherein the separate pending queues are prioritized such that the higher priority pending queues have a higher number of maximum entries than the lower priority pending queues.

5 ~~44~~. (New claim) The data processing system according to claim 1 ~~39~~, wherein the one or more requesting processors generate the processor requests over a first bus, and wherein the peripheral devices accept the processor requests over a second bus that has a different bus bandwidth from the first bus.

6 ~~45~~. (New claim) The data processing system according to claim 1 ~~39~~, wherein the controller marks a processor request as outstanding, when a corresponding peripheral device accepts such processor request.

A1

7 ~~46~~. (New claim) The data processing system according to claim 6 ~~45~~, wherein the controller places the processor requests in a return queue, after the peripheral devices respond to such outstanding processor requests.

2nd C3 47. (New claim) The data processing system according to claim 39 further including a shared memory device, wherein the controller generates the pending queues from an allocated free pool of entries on the shared memory device.

25

48. (New claim) The data processing system according to claim 47, wherein the controller variably adds entries to the pending queues from the allocated free pool of entries, only after the processor requests are generated.

49. (New claim) The data processing system according to claim 47, wherein the entries include pointers that point to memory locations on the shared memory device.

50. (New claim) The data processing system according to claim 47, wherein the processor requests include control information, addresses and data, and wherein the shared memory device is partitioned for storing the address and control information in a first memory array and for storing the data in a second separate memory array.

51. (New claim) A data processing system comprising:

one or more requesting processors that generate non-blocking processor requests directed to one or more peripheral devices;

a plurality of peripheral devices that accept the non-blocking processor requests;

a shared memory device; and

a memory controller responsive to the non-blocking processor requests that creates a plurality of separate pending queues on the shared memory device

corresponding to each one of the plurality of peripheral devices for queuing the non-blocking processor requests directed to a particular peripheral device in entries of a corresponding separate pending queue, wherein at least two separate peripheral devices process the non-blocking processor requests simultaneously, after retrieving such non-blocking processor requests from their respective separate pending queues.

11 10
~~52~~. (New claim) The data processing system according to claim ~~51~~, wherein the one or more requesting processors generate the non-blocking processor requests over a first bus, and wherein the peripheral devices accept the non-blocking processor requests over a second bus that has a different bus bandwidth from the first bus.

12 10
~~53~~. (New claim) The data processing system according to claim ~~51~~, wherein the non-blocking processor requests are prioritized in the pending queues such that the higher priority non-blocking processor requests are processed before the lower priority non-blocking processor requests.

13 10
~~54~~. (New claim) The data processing system according to claim ~~51~~, wherein the pending queues are prioritized such that the higher priority pending queues have a higher number of maximum entries than the lower priority pending queues.

55. (New claim) The data processing system according to claim 51, wherein the memory controller marks a non-blocking processor request as outstanding, when a corresponding peripheral device accepts such non-blocking processor request.

56. (New claim) The data processing system according to claim 55, wherein the memory controller places the accepted non-blocking processor requests in a return queue located on the shared memory device, after the peripheral devices respond to such outstanding non-blocking processor requests.

57. (New claim) The data processing system according to claim 51, wherein the memory controller generates the pending queues from an allocated free pool of entries on the shared memory device.

58. (New claim) The data processing system according to claim 57, wherein the memory controller variably adds entries to the pending queues, only after the non-blocking processor requests are generated.

59. (New claim) The data processing system according to claim 51, wherein the entries include pointers that point to memory locations on the shared memory device.

60. (New claim) The data processing system according to claim 51, wherein the non-blocking processor requests include control information, addresses and data, and wherein the shared memory device is partitioned for storing the address and control information in a first memory array and for storing the data in a second separate memory array.

61. (New claim) A method for processing data comprising the steps of:

generating processor requests that are directed to a plurality of peripheral devices by one or more requesting processors;

creating a plurality of separate pending queues that correspond to each one of the plurality of peripheral devices, for queuing the processor requests directed to a particular peripheral device in entries of a corresponding pending queue; and

processing two separate processor requests directed to corresponding peripheral devices simultaneously, after retrieving such processor requests from their respective separate pending queues.

62. (New claim) The method for processing data according to claim 61 further including performing dependency checking on the processor requests, to generate non-blocking processor requests.

25 24 63. (New claim) The method for processing data according to claim 62 20 19 further including generating the non-blocking processor requests by running real-time processes.

21 20 64. (New claim) The method for processing data according to claim 61 19 18 further including prioritizing the processor requests in the pending queues such that the higher priority processor requests are processed before the lower priority processor requests.

22 21 65. (New claim) The method for processing data according to claim 61 19 18 further including prioritizing the separate pending queues such that the higher priority pending queues have a higher number of maximum entries than the lower priority pending queues.

A1

23 22 66. (New claim) The method for processing data according to claim 61 19 18 further including variably adding entries to the pending queues, only after processor requests directed to corresponding peripheral devices are generated.

24 23 67. (New claim) The method for processing data according to claim 61 19 18 further including marking a processor request as outstanding after such processor request is accepted by a corresponding peripheral device.

30

68. (New claim) The method for processing data according to claim 61 further including:

allocating a shared memory space for entries of the separate pending queues; and

freeing the entries of the processor requests in the pending queues, after such processor requests are accepted by the peripheral devices.

69. (New claim) The method for processing data according to claim 68 further including placing the outstanding processor requests in the entries of a return queue on the shared memory space, after the corresponding peripheral devices respond to such outstanding processor requests.

A1

Sub C6

70. (New claim) The method for processing data according to claim 61 further including freeing entries of the processor requests in the return queue, after transmitting corresponding responses from the peripheral devices to the requesting processors.

71. (New claim) A system for processing data comprising:

means for generating processor requests that are directed to a plurality of peripheral devices by one or more requesting processors;

means for creating a plurality of separate pending queues that correspond to each one of the plurality of peripheral devices, for queuing the processor

requests directed to a particular peripheral device in entries of a corresponding pending queue; and

means for processing at least two separate processor requests directed to corresponding peripheral devices simultaneously, after retrieving such processor requests from their respective separate pending queues.

72. (New claim) The method for processing data according to claim 71 further including means for performing dependency checking on the processor requests, to generate non-blocking processor requests.

73. (New claim) The system for processing data according to claim 72 further including means for generating the non-blocking processor requests by running real-time processes.

74. (New claim) The system for processing data according to claim 71 further including means for prioritizing the processor requests in the pending queues such that the higher priority processor requests are processed before the lower priority processor requests.

75. (New claim) The system for processing data according to claim 71 further including means for prioritizing the separate pending queues such that the higher

priority pending queues have a higher number of maximum entries than the lower priority pending queues.

76. (New claim) The system for processing data according to claim 71 further including means for variably adding entries to the pending queues, only if processor requests directed to corresponding peripheral devices are generated.

77. (New claim) The system for processing data according to claim 71 further including means for marking a processor request as outstanding after such processor request is accepted by a corresponding peripheral device.

78. (New claim) The system for processing data according to claim 71 further including:

means for allocating a shared memory space for entries of the separate pending queues; and

means for freeing the entries of the processor requests in the pending queues, after such processor requests are accepted by the peripheral devices.

79. (New claim) The system for processing data according to claim 78 further including means for placing the outstanding processor request in the entries of a return queue on the shared memory space, after the corresponding peripheral devices respond to such outstanding processor requests.